|  |
| --- |
| Dept. of CSE, Pusan National University |
| UTF-8을 사용한 다국어 ID를 지원하는 메일서버 구현 중간보고서 |
| 소프트웨어시스템(김 경 석 교수님) |
|  |
| 권 상혁(200624410) 송 영찬(201024575) 염 재걸(201024576) 윤 정용(201024578) Team Name : 네 얼간이(4 idiots) |
| **2013-08-02** |

|  |
| --- |
| 이 문서는 2013년도 졸업과제 주제인 UTF-8을 사용하여 다국어 ID를 지원하는 메일서버를 구현에 관하여 중간기간까지의 설계상황 및 상세 내용을 담고 있습니다. |

UTF-8을 사용한 다국어 ID를 지원하는 메일서버 구현 중간보고서

**팀 명 : 네 얼간이**

**권 상혁(200624410) 송 영찬(201024575) 염 재걸(201024576) 윤 정용(201024578) Team Name : 네 얼간이(4 idiots)**

Dept. of CSE, Pusan National University

2013년 8월 2일

**요 약**

이 문서는 2013년도 졸업과제 주제인 UTF-8을 사용하여 다국어 ID를 지원하는 메일서버를 구현에 관하여 중간기간까지의 설계상황 및 상세 내용을 담고 있습니다.

주제어: UTF-8, Mail Server, ASCII, RFC 규격, SMTP, James …

목차

1. 요구조건 및 제약 사항 분석에 대한 수정사항--------------------2
2. 요구조건
3. 제약사항
4. 수정사항
5. 설게 상세화 및 변경 내역----------------------------------2
6. 변경 내역
7. 상세 설계내역
8. 갱신된 과제 추진 계획------------------------------------11
9. 수정된 개발 일정
10. 구성원 별 진척도----------------------------------------12
11. 현재 개발 진척도
12. 보고 시점까지의 과제 수행 내용 및 중간결과 ------------------12
13. 중간시점 과제 수행내용
14. 중간결과
15. 요구조건 및 제약 사항 분석에 대한 수정사항
16. 요구조건

* 메일 전송 및 수신을 위한 메일서버 환경 구현
* 사용자의 계정 및 메일서버를 DB와 연동하여 관리할 수 있게 구현
* 메일을 보내고 받을 수 있는 WebMail Page를 구현
* 사용자의 ID를 UTF-8을 사용하여 한글 및 다국어지원 이 가능하도록 구현

1. 제약사항
   * + UTF–8을 사용해서 다국어 지원이 가능해야 하므로 Punycode사용은 주제에 부적합
     + Domain파트는 따로 encoding을 할 필요 없음
     + 메일서버를 Open Source 를 참조하여 구현하는 것이 간단할 수도 있다
     + 개인의 PC를 메일서버로 사용하게 되고, PC간 메일 전송이 가능해야 한다
2. 수정사항

* 착수보고서에서 제시되었던 Downgrade 및 개발환경에 대한 내용을 수정하였고, 여러 번의 시행착오 끝에 Windows 환경에서 Java 및 JSP를 사용하여 해당 주제의 구현을 진행하였다.
* 초기에 계획하였던 Android application을 구현하는 것 보다, WebMail 페이지를 구현하여 시간을 들이는 것이 더욱 완성도를 높일 수 있겠다는 판단 하에 Android관련 내용은 삭제하였다.

1. 설게 상세화 및 변경 내역
2. 변경 내역

* 개발환경 변경내역
* 운영체제 : Windows 7 & Windows XP
* 개발언어 : Java, JSP
* 구현환경 : Eclipse
* 메일서버 : James 2.3.2
* 컴파일러 : Java Compiler
* Database : MySql 5.6
* Linux 운영체제를 사용해서 구현을 하려다 Windows 환경으로 변경
* 구현 language 변경 ( Python -> Java)
* Sendmail 구현 과정에서 OpenSource인 James 2.3.2 메일서버 사용으로 변경

1. 상세 설계내역

**James Mail Server**

|  |
| --- |
| SMTP(send)  SMTP(send)  **내 ISP’s**  **메일서버**  Internet  **내 컴퓨터**  **(Client)**  **Internet**  POP3(receive)  **메일서버**  POP3(receive)  그림 1. |

ISP’s 메일서버 부분이 SMTP프로토콜을 해석 할 수 있는 중간서버 역할을 하게 되는데, 이에 해당하는 서버가 James로 사용가능 하다.

James를 사용하여 자신의 PC를 메일서버로 구축하기 위해서는 우선적으로 Java가 설치되어 있어야 사용이 가능하였고, James 서버의 환경설정은 config.xml 파일을 통해서 제어 할 수 있었기 때문에 이 문서의 수정이 필요하다.

|  |
| --- |
| 그림 2 config.xml 파일에 IP주소 추가 |

위와 같이 파일을 수정 한 뒤에 James를 구동시켜보면 그림3과 같은 화면이 출력이 된다면 정상적으로 메일서버가 동작한다는 의미이다.

|  |
| --- |
| 그림3 메일서버 정상동작 화면 |
| **MySql연동**  앞서 구현했던 James를 MySql과 연동하여 DB에서 사용자관리 및 메일파일을 관리하기 위해서는 James에서 메일을 구축할 때, config.xml파일을 수정했던 것에서 추가적으로 문서의 내용을 수정 해 주어야 한다.  기존의 James환경에서는 메일을 보내고 받으면 해당 파일이 파일로 관리되기 때문에, 파일을 관리하기 까다롭기 때문에 Mysql을 사용하여 DB를 통해 관리한다면 더욱 편리하게 메일서버를 관리 할 수 있다.  Mysql을 5.6을 설치하는 과정에서 계정을 추가해주고 이계정을 통해서 config.xml문서를 수정할때 사용하도록 한다.   |  | | --- | | 그림 4 path변경 |   위의 그림 4 와 같이 기존에 Path가 <file://var> .....로 설정되어 있는 것들을 아래에 추가한 코드인db://maildb…..와 같이 바꿔주게 되면 기존에 메일파일이 저장되는 방식이 DB의 경로에 저장이 되게된다. 저 부분 외에도 <file://var>....경로로 설정되어 있는 코드들을 모두 주석처리 또는 삭제를 하고 경로가 db://maildb/,,,,로 향하도록 설정해 주면 된다.  그리고 MySql과 연동하기 위해서 config.xml 파일에 추가적으로 작성해야 하는 소스코드는 아래와 같다.   |  | | --- | | <data-source name="maildb" class="org.apache.james.util.dbcp.JdbcDataSource">  <driver>com.mysql.jdbc.Driver</driver>  <dburl>jdbc:mysql://127.0.0.1/test?autoReconnect=true</dburl>//생성한 DB명 경로 설정  <user>yjg</user>//DB설치시 추가한 계정명  <password>yjg1004</password>//DB설치시 추가한 계정의 Password  <max>20</max>  </data-source> |   위 소스코드는 DB와 연동하기 위해서 필요한 라이브러리를 사용하는 내용을 포함하고 있다. 그리고 3번재 line의 ip주소 다음의 test는 Mysql에서 생성한 Database의 이름이다. User와 password는 초기 Mysql설치 시 설정하였단 계정과 password를 입력해서 DB에 접근이 가능하도록 구현하였다.  **메일수신 JSP**  메일 서버 웹 페이지(JSP)에서 서버에 등록된 사용자에 대한 메일 리스트를 출력하는 기능을 구현하였다. 현재까지는 단순히 리스트를 글자로 뿌리는 것 까지만 구현하였으며, 추후 웹 페이지 양식을 꾸며 표시할 계획이다.  계정 생성시에 계정의 ID를 UTF-8로 인코딩 된 형식으로 저장하며, 화면에 보여줄 때는 이를 다시 디코딩하여 보여준다. 인코딩 및 디코딩에 대한 부분은 보고서 내의 다른 부분에서 다루기로 함. 구체적으로 메일 리스트를 표현하는 부분은 코드를 통해 설명하겠다  Properties properties = **new** Properties();  properties.put("mail.pop3.host",mailPop3Host);  Session emailSession = Session.getDefaultInstance(properties);  POP3Store emailStore = (POP3Store) emailSession.getStore(mailStoreType);  emailStore.connect(mailUser, mailPassword);  위의 코드를 통하여 POP3의 편지함에 접속하였으므로 POP3 서버 내의 받은 메일들이 저장된 공간 (여기서는 Folder를 사용함.)을 열어야 한다. 그에 해당하는 코드는 아래와 같다.  Folder emailFolder = emailStore.getFolder("INBOX");  emailFolder.open(Folder.READ\_ONLY);  POP3 서버 내의 메일함에 접근하였으므로, 메일의 내용을 모두 뽑아오는 코드를 작성하면 된다. 코드는 아래와 같다.  Message[] messages = emailFolder.getMessages();  **for** (**int** i = 0; i < messages.length; i++) {  Message message = messages[i];  out.println("Subject: " + message.getSubject()+"<br>");  out.println("Content Type: "+ message.getContentType()+"<br>");  out.println("From: " + URLDecoder.decode(message.getFrom()[0].toString(),"UTF-8")+"<br>");  out.println("Text: "+ message.getContent().toString()+"<br><br>");  }  위의 예시에서는 발신인, 제목, Content-Type 및 내용만을 가져오는 것으로 작성이 되어 있으나, message의 get- 메소드를 통해 메일의 다른 내용들도 가져올 수 있다. 여기서 Message 객체는 E-Mail 메시지를 표현한 것으로, E-mail 양식에 필요한 모든 내용이 들어있다.  아래는 간단한 테스트를 통해 해당 계정에 알맞는 E-mail 리스트를 출력하는 과정이다.    그림 5 이 화면을 통해 계정에 로그인    그림 6 해당 계정 (gggg2)이 받은 메일함이 출력된 화면.  **MailClient의 SendMail Part 구현**   1. MailForm 구현   C:\Documents and Settings\Administrator\바탕 화면\MailSend.JPG  그림 7 보내는 사람과 받는 사람, 제목, 내용을 입력받는 HTML Form  C:\Documents and Settings\Administrator\바탕 화면\Finish_MailSend.JPG  그림 8 메일이 정상적으로 전송되었을 경우의 메시지   1. 소스코드     그림 9 Mail From 소스코드  **Encoding/Decoding을 통한 UTF-8 계정 지원 기능**    SMTP 서버에 접속해 메일을 전송할수 있는 sendmail 클라이언트를 작성하였는데 javax.mail.internet 패키지를 사용하여 메일서버를 구축하였다  이 패키지를 이용하여 메일을 전송할 시에 local 파트에 아스키외에 문자, 즉 한글이 올경우 javax.mail.internet.addressexception 에러가 발생하여 메일 전송이 되질 않는다  이를 해결하기 위해 로컬과 도메인 파트를 따로 떼어낸 뒤에 로컬 파트를 자바에 포함되어있는 URL Encoder.encode를 사용해 UTF-8 형식으로 전환한뒤 다시 주소를 합쳐 메일을 전송하게된다  ex) 권상혁@localhost - > %EA%B6%8C%EC%83%81%ED%98%81@localhost  현재 인코딩한 메일주소를 포함해 메일을 서버에 보내는것 까지 구현이 되었다.  메일 서버에서는 한글계정명을 생성할시 한글을 위와 같은 방법으로 인코딩하여 DB에 저장하여 구현 할생각이며 서버는 인코딩된 메일주소를 받아 DB를 통해 검색한뒤 일치하는 계정의 메일 박스에 전송하게 되고 받는 사람의 인터페이스에서 다시 디코딩과정을 거쳐 유저 클라이언트의 인터페이스에서는 한글로 표시가 되게 구현할 예정이다. |

1. **갱신된 과제 추진 계획**
2. 수정된 개발 일정

|  |  |
| --- | --- |
| 개발 기간 | 개발 내용 |
| **6/24~ 6/30** | 구현에 필요한 기초 지식 습득 및 역할 세부분담 |
| **7/1 ~ 7/ 7** | James SMTP 서버 통신 부분 구현 |
| **7 / 8~ 7 / 14** | James SMTP Server / Client 연동 |
| **7 /15 ~ 7 /21** | 메일서버 Test 및 DB연동 |
| **7/ 22 ~ 7/ 28** | JSP를 사용한 Web Mail 페이지 구현 |
| **7 /29~ 8/ 1** | 중간보고서 작성 및 검토 |
| **8/2** | 중간보고서 제출 |
| **8/ 3~ 8/9** | Web Mail Test 및 Mail Server 동작 Test |
| **8/10 ~8/23** | UTF-8 decoding/encoding 부분 설계 |
| **8/24 ~ 8/30** | 전체적인 Sever 점검 및 Test |
| **8/ 31 ~9/ 13** | 메일 시스템 점검 |
| **9/14 ~ 9 /26** | 최종 점검 및 최종보고서 작성 |
| **9/27** | 최종보고서 제출 |

1. **구성원 별 진척도**
2. 현재 개발 진척도

|  |  |  |
| --- | --- | --- |
| **구성원** | 역 할 | 공통 역할 |
| **송 영찬** | **Java 및 JSP를 사용하여 Web페이지에서 메일**  **송신기능 구현** | **James – 메일 서버환경 구현**  **보고서 수정 및 검토** |
| **염 재걸** | **구현된 메일서버를 MYSQL(데이터베이스)에 연동 및 TEST** |
| **윤 정용** | **Java,JSP를 사용하여 Web페이지에서 수신 메일함 구현 및 테스트** |
| **권 상혁** | **Encoding/Decoding을 통한 UTF-8 계정 지원 기능** |

1. **보고 시점까지의 과제 수행 내용 및 중간결과**
2. 중간시점 과제 수행내용

|  |
| --- |
| 1. telnet으로 들어가서 syc 라는 user를 추가 |
| 1. DB에 테이블 및 해당 유저가 추가 되었음을 확인 |
| 1. Webmail 페이지를 통해서 메일 송신 및 정상동작 확인   C:\Documents and Settings\Administrator\바탕 화면\MailSend.JPG  C:\Documents and Settings\Administrator\바탕 화면\Finish_MailSend.JPG |
| 1. DB에 해당 메일이 저장 되었음을 확인 |

1. 중간결과

* WIndows환경에서 java의 오픈소스 메일서버인 james 2.3.2를 통해서 메일서버 구축
* MySql(DB)와의 연동을 통해서 송수신한 메일의 데이터를 효과적으로 관리할 수 있도록 구현
* JSP를 사용하여 메일을 송수신 할 수 있는 Form을 구현하여 Test
* 추가적으로 UTF-8 encoding 및 decoding 과정을 대략적으로 구현하였지만 개발 진행중에 있으므로 아직 미test단계이다.